Rumor Source Finder

Abhishek Kumar Srivastava, Deepak Gupta

Department of Computer Science
North Carolina State University, Raleigh, North Carolina, USA
Email: {asrivas3, dgupta22 }@ncsu.edu

***Abstract* —** In this project, we are considering the problem of finding the rumor source in a network based on the nodes that are infected at any particular time. In the past various researches has been done in this field by calculating the Rumor Centrality in SIR model and its variant SI model [2]. In this project, we have modeled the rumor spreading in the network using SI (Susceptible-Infected) model and then constructed a Maximum Likelihood estimator using rumor centrality for finding the rumor source. The modeling and simulation are done is python and results of the accuracy are obtained on a sample graph from Facebook.

Keywords – Rumor Finder, Rumor Centrality, SI (Susceptible-Infected) model, SIR (susceptible-infected-recovered) model, Maximum Likelihood Estimator, Modeling and Simulation.

I. INTRODUCTION

Rumor propagation and source detection are one of the most important challenges in the world today. Finding a rumor source is useful in many scenarios like social network analysis, privacy, cybersecurity, epidemiology and information assurance [3]. In these scenarios, detecting rumor source is of interest as this source may be malicious information trying to defame someone, computer virus, patient zero or an influential person.

There are many key challenges to rumor source detection like what is the underlying topology, how to gather relationships in form of a graph, how to construct and represent such a graph, how accurate is the information represented in the graph, what is the error distribution and how to construct the rumor source estimator.

Although there are many works discussing the above-mentioned challenges, our work focus on how we can actually construct and implement a rumor source estimator. Here we consider the problem of rumor source detection when given a snapshot of the state of the network at an unknown point in time. The graph may have some infected and healthy nodes. The nodes affected by the rumor are referred to as infected while those which are not infected are referred to as healthy nodes. The work focus on estimating the probability of a node as a true source spreading the rumor from the given infected graph, where estimator finds the node with maximal rumor centrality via maximum likelihood estimation in a graph with generic spreading under the SI model.

Similar problems and approaches have been considered previously. In [1], the paper presents the approach for maximum likelihood estimation of a single source of a rumor in a graph where rumor propagates according to the SI model.

II. RELATED WORK

Various research works have been done in the past on finding the source of rumor these were primarily based on viral epidemics spread in the population [1-3]. These were primarily based on the natural model of viral epidemic known as SIR (susceptible-infected-recovered) model. In this model, there are three type of nodes:

1) **Susceptible nodes** – these are the nodes that are susceptible to be affected by a rumor/infection;
2) **Infected nodes** – these are the nodes that are already infected from rumor/infection and have not recovered yet;
3) **Recovered nodes** – these are the nodes that were infected in the past but have currently recovered from the infection.

These research works have provided us with the understanding of how the structure of network and rate of infection affects the spread of the epidemic in the populous. But still, these researches have not been very successful in predicting the source of an infection or rumor. This is primarily due to the sheer complexity of the networks that makes the correct prediction quite challenging.

In this project we are using a variant of SIR model called SI (susceptible-infected) model, to reduce the complexity of calculation and prediction. In this model there are two types of nodes:

a) Susceptible nodes – these are the nodes that are susceptible to being infected;
b) Infected nodes – these are the nodes that are infected with the rumor/infection and will remain infected.

In this project, we describe how to construct a graph. Then using a node as the source, how a rumor is propagated into the network following Breadth First Search (BFS) pattern in the graph. And then how to calculate the rumor centrality for the infected nodes and predict the source of the rumor using the rumor centrality values of the nodes.

III. METHODOLOGY

A. Rumor Spreading Model

We consider a network of nodes modeled as an undirected graph G(V, E), where V is the number of nodes in the graph and E is the number of edges in the graph. We consider the case where only one node is the rumor source. We then use the SI model (Susceptible-Infected model) for the rumor spreading which does not allow for any node to recover from the rumor. Thus, once a node is infected, it remains infected forever. But it can spread the rumor to susceptible nodes. Once node i is infected, it can infect node j if and only if an edge (i, j) ϵ E exists between them.

B. Rumor Source Estimator: Maximum Likelihood

If a rumor starts from source 's' and spreads in network G, thereby infecting N nodes. Then by definition, G_N must be connected sub-graph of G containing only infected nodes. Our goal is to predict the true source of the rumor. However, a priori we do not know from which source the rumor started. Therefore, we shall assume a uniform prior probability of the source node among all nodes of G_N. Therefore, by definition Maximum Likelihood estimator is:

$$\hat{v} \in \arg\max_{v \in G_N} P(G_N|v)$$

where $\hat{v}$ is the estimated source and $P(G_N|v)$ is the probability that G_N is the observed graph given v is the source of the rumor. Hence, we will estimate $P(G_N|v)$ for all v $\in$ G_N and will select that v as rumor source whose probability is maximum.

C. Rumor Centrality: Definition

The quantity R(v,G_N) counts the number of distinct ways a rumor can spread in the network G_N starting at node 'v'. This quantity is directly proportional to $P(G_N|v)$[1]. This assigns a non-negative score to each node in G_N, this score is called Rumor Centrality of the node. The node with maximum score is called the rumor center of the network. The rumor center is the ML estimation of the rumor source for the graph.

D. Calculating Rumor Centrality

Let T_u^v be the number of nodes in the subtree rooted at node u with node v as source.

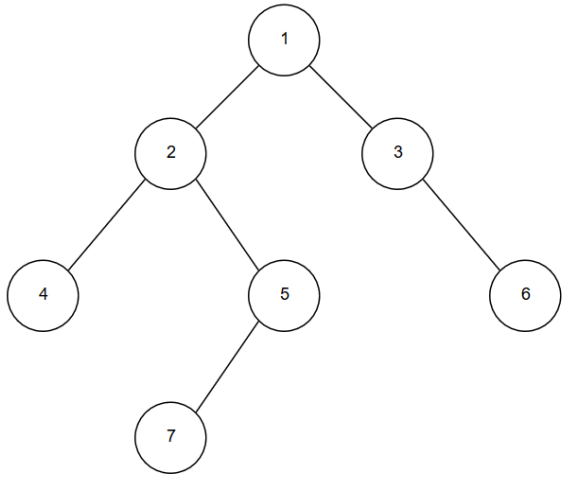


Figure: 1

For example, in the Figure 1, if we consider the graph node 1 as source and subtree rooted at node 2, then T_2^1 = 4. Similarly, $T_3^1 = 2$.

We can count the permitted permutations of G_N with v as a source. Since we have N nodes in G_N and first position is fixed with v as a source. Therefore, we need to fill the rest of N-1 slots with other nodes.

The basic constraint is that a parent will be infected before its children. Therefore, in the permutation, parent node will come before the child nodes. This gives a recursive relation between rumor centrality $R(v, GN)$ and the rumor centrality of its immediate child's $R(u, T_u^v)$, with $u \in child(v)$. This leads to the following relation:

$$R(v, G_N) = (N-1)! \prod_{u \in child(v)} \frac{R(u, T_u^v)}{T_u^v!}$$

Since a leaf node 'l' will have 1 node and 1 permitted permutation, so $R(l, T_l^v)$ = 1. Thus, if the recursion is continued until leaf of the tree is reached, then the number of permutations for graph G_N rooted at 'v' will be given as:

$$R(v, G_N) = N! \prod_{u \in G_N} \frac{1}{T_u^v} \quad \text{.....(1)}$$

For example, in figure 1. the rumor centrality for Node 1 will be

$$R(1, G_N) = 7! * \left\{\frac{1}{1*1*2*4*7*2*1}\right\}$$
$$R(1, G_N) = 45$$

IV. IMPLEMENTATION

The rumor centrality of each node on graph G_N is calculated by using message passing algorithm. If 'u' and 'v' are two neighbor nodes, then their rumor centrality can be calculated as [2]:

$$R(u, G_N) = R(v, G_N) \frac{T_u^v}{N - T_u^v} \quad \ldots\ldots\ldots(2)$$

First, any node v is selected as source and then the messages are passed between parent and children in two forms:

- up messages
- down messages

In up messages, a child node calculates the size of its subtree and the cumulative product of the size of the subtrees of all nodes in its subtree. It then passes these values to the parent. The parent then uses these up messages to calculate the size of its own subtree and its cumulative subtree product. These messages are exchanged between a child and its parent until the source node is reached. The source node can then calculate its rumor centrality by using equation (1).

The source node then passes its calculated rumor centrality in downward fashion via down messages to its children. The children calculate their own rumor centrality using equation (2). These down messages are exchanged until leaf nodes are reached.

Thus, we can get rumor centrality for each node in G_N in O(N) time complexity. The node with the maximum rumor centrality will be declared as the rumor center.

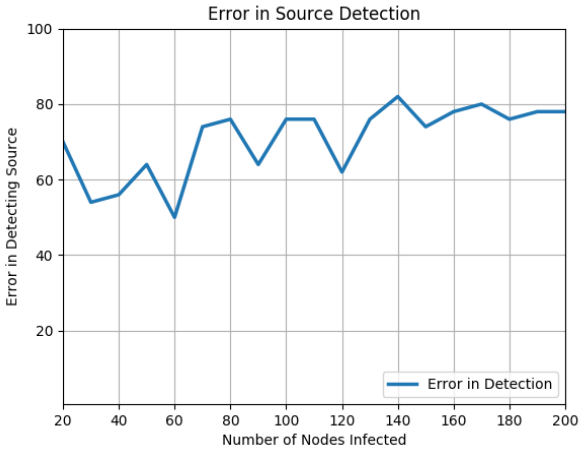
The full algorithm is as follows [2]:

Algorithm 1 Rumor Centrality Message-Passing Algorithm

Choose a root node $v \in G_N$
for u in G_N **do**
  if u is a leaf **then**
    $t^{up}_{u \to \text{parent}(u)} = 1$
    $p^{up}_{u \to \text{parent}(u)} = 1$
  else
    if u is root v **then**
      $\forall\, v' \in \text{child}(v)$: $r^{down}_{v \to v'} = \dfrac{N!}{N \prod_{j \in \text{child}(v)} p^{up}_{j \to v}}$
    else
      $t^{up}_{u \to \text{parent}(u)} = \sum_{j \in \text{child}(u)} t^{up}_{j \to u} + 1$
      $p^{up}_{u \to \text{parent}(u)} = t^{up}_{u \to \text{parent}(u)} \prod_{j \in \text{child}(u)} p^{up}_{j \to u}$
      $\forall\, u' \in \text{child}(u)$: $r^{down}_{u \to u'} = r^{down}_{\text{parent}(u) \to u} \frac{t^{up}_{u \to \text{parent}(u)}}{N - t^{up}_{u \to \text{parent}(u)}}$
    end if
  end if
end for

V. RESULTS

A sample network from Facebook is taken with 5000 number of nodes. A variable number of nodes are infected ranging from 20 to 200. For each number of infected nodes, the rumor centrality algorithm is run for a fixed number of iterations. In each iteration, the result from the algorithm is compared with the actual rumor source to find out the success rate. The following graph shows the results of the simulation:



The graph clearly shows that as the number of infected nodes increases in the network, the accuracy of predicting the correct rumor source decreases.

REFERENCES

[1] D. Shah and T. Zaman. Rumor centrality: A universal source detector. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 199–210, New York, NY, USA, 2012. ACM.

[2] D. Shah and T. Zaman, "Rumors in a network: Who's the culprit?" IEEE TIT, vol. 57, no. 8, pp. 5163-5181, 2011.

[3] S. Spencer and R. Srikant. On the impossibility of localizing multiple rumor sources in a line graph. ACM SIGMETRICS Performance Evaluation Review, 43(2):66–68, 2015.